

ASSIGNMENT 2: I/O ELEVATORS

CS 444: OPERATING SYSTEMS II

Fall 2016

Abstract

This project involves researching and implementing the LOOK or CLOOK i/o elevators. The goal is to gain a nd understanding of how elevators work inside of an operating system, how they are implemented, and how to debug them. Through work our process of learning and implementing the ioscheduler is shown.

Joshua D. Bowen

Chris J. Mendez

October 24, 2016

1 SSTF Design

Our SSTF design basically consists of keeping track of the head of the queue. We consistently move down the list from one item to the next until it reaches the end. Because the linked list is circular in nature we opted for CLOOK as CLOOK means that we immediately reset to the head which a circular linked list makes easy as it's only one additional step forward. As each item is reached it's added to another queue to be dispatched. The dispatcher checks to make sure the io requests are proper and then completes them and removes them from the circular linked list.

2 Questions

2.1 What do you think the main point of this assignment is?

The main point of the assignment was to become familiar with the way files, in this case specifically io elevators, are loaded and processed by the kernel. We got to go hands on with the actual files as part of the kernel and become familiar with how they are structured and written. Additionally we also supposed to become familiar with the qemu man pages and in a way the entire concept of referencing and reading documentation for a large complicated structure like the yocto.

2.2 How did you personally approach the problem? Design decisions, algorithm, etc.

We started by doing research into elevator algorithms and specifically into the LOOK and CLOOK elevators. Due to the circular nature of linked lists in the kernel we decided that CLOOK would be the better choice. We decided this because CLOOK wants you to reset to the front of the queue every time, and seeing as the lists are circular in nature this took no extra work beyond what was already happening by traversing down the list.

Once we had decided that this was the version of LOOK we wanted to implement we went into the yocto files and found noop-iosched.c. Using noop-iosched.c we created a sstf-iosched.c. Using noop-iosched we familiarized ourselves with how an elevator scheduler was written inside of the kernel. We could see all of the functions that would need to be modified. From there we looked at each function individually and assessed the changes that would need to be made to each to make the scheduler proper.

2.3 How did you ensure your solution was correct? Testing details, for instance.

We ensured our solution was correct by utilizing kprint and dmessage. Whenever an io process was a read we would also print the content to the terminal. This allowed us to get an idea of where the scheduler was in the list and let us verify that it was acting as it should be. We verify that it's acting as it should by seeing that the input is getting progressively smaller and then jumps back to a large number.

2.4 What did you learn?

We learned about different elevator functions that are used to handle io, as well as what a lot of the different qemu flags do in relation to the activating of the OS.

3 History

3.1 Git Version Control Log

Commit	Message
commit a64c8cf3faf3a2d401c338915d0bf20490664867 (HEAD, master) Author: bowenjos jbowenjos@oregonstate.edu Date: Mon Oct 24 15:49:10 2016 -0700	sstf changes 10/24
commit dc4d711475e92d7481ce2bc8bc4d4dc965b05940 Merge: d924130 f28ea8a Author: Mendezc1 jmendezc@oregonstate.edu Date: Mon Oct 24 13:36:08 2016 -0700	Merge branch 'master' of https://github.com/bowenjos/cs444-020
commit d924130862dbd358a3bb156ea7632fcf0c26bce8 Author: Mendezc1 jmendezc@oregonstate.edu Date: Mon Oct 24 13:36:04 2016 -0700	attempting patch
commit f28ea8a473ea0bec0c156de092cbda96fada9f33 (origin/-master, origin/HEAD) Merge: 713d43c d839fdc Author: Chris Mendez jmendezc@oregonstate.edu Date: Sun Oct 23 20:53:36 2016 -0700	Merge branch 'master' of https://github.com/bowenjos/cs444-020
commit 713d43ca62592180a2356cfcc985e5f5e0a414d9 Author: Chris Mendez jmendezc@oregonstate.edu Date: Sun Oct 23 20:53:29 2016 -0700	Adding tex file
commit d839fdc6c6952ef7c0bc53180b0dfef7c4892597 Author: Mendezc1 jmendezc@oregonstate.edu Date: Sat Oct 22 11:58:31 2016 -0700	attempting to improve iosched to be more clook instead of noop
commit f369035454f15b7e6c84e4bfee60e99e6da46987 Author: Mendezc1 jmendezc@oregonstate.edu Date: Sat Oct 22 10:25:24 2016 -0700	sstf-iosched.c added

3.2 Work Log

Date	Time	What
10/22/2016	10am-1pm	Started assignment, familiarized ourselves with the noop-iosched.c file as well as the LOOK and CLOOK functions. Attempted to load the qemu without virtio and eventually got it working. Ran into an issue with yocto running with our updates.
10/23/2016	12am-10pm	Got questions answered by Kevin, we were on the right path however for some reason the yocto was continuing to load the generic version and not our version. As soon as we get that figured out we can work on making sure our implementation of sstf-iosched is working properly. Began working on write up and setting the latex file up.
10/24/2016	3pm-4pm	Met in the library to sort out our bugs. We managed to get the qemu to use our version of the yocto but were met with crashes. We determined that the crashes were likely the result of our iosched and went to work fixing the implementation.